

# **PERANCANGAN SISTEM PENGENALAN JENIS TANAMAN OBAT DENGAN KAMERA BERBASIS ANDROID**



**Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada  
Program Studi informatika Fakultas Komunikasi dan Informatika**

**Oleh:**

**WIDHAR DWIATMOKO**

**L200160179**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2020**

**HALAMAN PERSETUJUAN**

**PERANCANGAN SISTEM PENGENALAN JENIS TANAMAN OBAT  
DENGAN KAMERA BERBASIS ANDROID**

**PUBLIKASI ILMIAH**

oleh:

**WIDHAR DWIATMOKO**  
**L200160179**

Telah diperiksa dan disetujui untuk diuji oleh:  
Dosen Pembimbing



**Ir. Bana Handaga, M.T. PhD**

**NIK.793**

## HALAMAN PENGESAHAN

# PERANCANGAN SISTEM PENGENALAN JENIS TANAMAN OBAT DENGAN KAMERA BERBASIS ANDROID

OLEH

WIDHAR DWIATMOKO

L200160179

Telah dipertahankan di depan Dewan Penguji  
Fakultas Komunikasi dan Informatika  
Universitas Muhammadiyah Surakarta  
Pada hari Senin, 20 April 2020  
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. Ir. Bana Handaga, M.T. PhD

( ..... )

(Ketua Dewan Penguji)

2. Dr. Endah Sudarmilah, S.T., M.Eng

( ..... )

(Anggota I Dewan Penguji)

3. Devi Afriyanti Puspa Putri, S.Kom., M.Sc( ..... )

(Anggota II Dewan Penguji)

Digital dengan CamScanner

Dekan  
Fakultas Komunikasi dan Informatika



Nurgiyatna, S.T., M.Sc., Ph.D.

881

## PERNYATAAN

Dengan ini saya menyatakan bahwa dalam naskah publikasi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 20 April 2020

Penulis



**WIDHAR DWIATMOKO**

**L200160179**

## **SURAT KETERANGAN LULUS PLAGIASI**

No Surat .....

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : Widhar Dwiatmoko  
NIM : **L200160179**  
Judul : **PERANCANGAN SISTEM PENGENALAN JENIS TANAMAN  
OBAT DENGAN KAMERA BERBASIS ANDROID**  
Program Studi : Informatika  
Status : **Lulus**

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

7 Submitted to University... 1%

# PERANCANGAN SISTEM PENGENALAN JENIS TANAMAN OBAT DENGAN KAMERA BERBASIS ANDROID

## Abstrak

Kekayaan alam hutan tropic Indonesia menyimpan berbagai potensi keanekaragaman hayati yang sangat melimpah. Salah satu hasil alam yang dapat dimanfaatkan dibidang Kesehatan sebagai obat herbal untuk penyembuhan penyakit. Tanpa disadari ternyata banyak tanaman herbal yang tumbuh disekitar rumah, namun banyak yang belum mengetahui dikarenakan mereka sulit diketahui dengan mata biasa. Berkat perkembangan Teknologi 4.0, penggunaan *smartphone* sudah menjadi hal umum. Perangkat *android* sudah bertransformasi untuk penunjang kehidupan dimasyarakat. Setiap perangkat *smartphone android* terkoneksi dengan kamera, namun kamera tersebut hanya digunakan untuk melakukan kegiatan fotografi. Kamera tersebut juga belum memiliki sistem khusus untuk melakukan proses pengenalan suatu objek. Tujuan dari penelitian ini ialah menerapkan kamera *smartphone* pada *android* agar bisa mengenali daun tanaman obat, sehingga mengembalikan animo masyarakat untuk memakai daun sebagai obat. Dengan metode *Deep Learning* dan arsitektur *CNN*, dapat membuat kamera yang sebelumnya tidak bisa mengenali suatu objek menjadi mampu mengenali objek sesuai data training. Penelitian menggunakan metode *Deep Learning* untuk membuat sebuah sistem pengenalan daun herbal dengan memanfaatkan kamera *smartphone android*. Hasil dari metode tersebut diperoleh 100% akurasi terhadap 6 jenis daun herbal yang telah dipotret diatas *background* putih.

**Kata Kunci:** *android*, *CNN*, daun , herbal ,*deep learning*, kamera.

## Abstract

The natural wealth of Indonesia's tropical forests holds a wealth of biological potential which is very abundant. One of the natural products that can be used in the field of health as herbal medicines for healing diseases. Without realizing it turns out many herbs that grow around the house, but many are not yet known because it is difficult to understand with the ordinary eye. Thanks to the development of Technology 4.0, smartphone usage has become common. Android devices have been transformed to support life in the community. Every android smartphone is connected to a camera, but the camera is only used for photography. This camera also does not have a special system for the process of recognizing an object. The purpose of this study is to apply a smartphone camera on Android in order to be able to recognize the leaves of medicinal plants, thereby restoring public interest in using leaves as medicine. With the Deep Learning method and CNN architecture, it can make a camera that previously could not discuss an object to be able to focus the object according to data training. Research using the Deep Learning method to create a herbal leaf recognition system using an android smartphone camera. The results of the method obtained are 100% accurate on 6 types of herbal leaves that have been photographed on a white background.

**Keywords:** *android*, camera, *CNN* ,*deep learning*, herbal, leaf.

## 1. PENDAHULUAN

Kekayaan alam hutan tropik Indonesia menyimpan berbagai potensi keanekaragaman hayati yang sangat melimpah (S. Ifandi, Jumari, & S. Widodo AS, 2015). Keanekaragaman hayati akan dapat dimanfaatkan kegunaanya diberbagai bidang, salah satunya dibidang kesehatan. Salah satu hasil alam

adalah tumbuh- tumbuhan yang digunakan sebagai obat untuk menyembuhkan berbagai penyakit (Sari *et al.*, 2015). Obat dari tumbuhan tentunya akan sangat minim efek samping, karena obat terbuat dari bahan-bahan organik. Namun, ketidaktahuan masyarakat untuk mengenali tumbuhan-tumbuhan yang dapat digunakan untuk obat alami. Maka masyarakat akan memilih cara mudah yaitu dengan membeli obat kimia dari apotek. Obat-obatan kimia mengandung bahan kimia yang bersifat anorganik dan murni, sementara tubuh manusia bersifat organik dan kompleks (Ni'mah, Sutojo and Setiadi, 2018). Obat kimia memang tidak dilarang untuk dikonsumsi, namun tidak baik untuk dikonsumsi dalam jangka Panjang. Maka dari itu dibutuhkan perubahan pola pikir masyarakat untuk kembali menggunakan obat herbal dibanding menggunakan obat kimia.

Perkembangan teknologi 4.0 di Indonesia akan berdampak pada kehidupan masyarakat. Teknologi melahirkan sistem yang mampu membantu permasalahan yang ada di masyarakat. Seperti Masalah yang dipaparkan diatas, maka dibutuhkan sistem yang mempermudah masyarakat untuk mengenali tanaman obat dengan cepat. Sistem lebih baik jika dapat digunakan secara *mobile* karena lokasi objek berada di luar ruangan. Kini *smartphone* sudah menjadi media komunikasi pokok (Gifary, 2015). Hal ini menjadikan *smartphone* menjadi perangkat *mobile* yang paling sering digunakan untuk kebutuhan sehari-hari. *Smartphone* kini telah dilengkapi kamera bawaan dengan ukuran pixel yang beragam. Lahirnya *Tensorflow* beberapa tahun terakhir sebagai *open source framework machine learning*, menjadikan banyak *developer* dapat mempelajari algoritma *learning process* dari *machine learning*. *Tensorflow* merupakan antarmuka untuk mengekspresikan algoritma pembelajaran mesin dan untuk mengeksekusi perintah dengan menggunakan informasi yang dimiliki tentang objek tersebut atau target yang dikenali serta dapat membedakan objek satu dengan objek lainnya (Santoso and Ariyanto, 2018).

Peneliti menggunakan kamera *smartphone* untuk menangkap citra daun dan dapat mengenalinya serta memberikan cara meramu daun tersebut menjadi tanaman obat. Sistem akan berjalan pada sistem operasi *android*, serta proses *training* dataset akan dilakukan menggunakan *open source tensorflow lite*. Alasan peneliti menggunakan sistem operasi *android* dikarenakan sistem android merupakan sistem operasi yang mampu berjalan secara *mobile*, alasan lain karena sistem operasi *android* banyak digunakan masyarakat sekarang. Dengan penelitian ini peneliti berharap agar masyarakat menggunakan tumbuhan herbal sebagai obat alternatif.

## 2. METODE

### 2.1. Arsitektur Neural Network

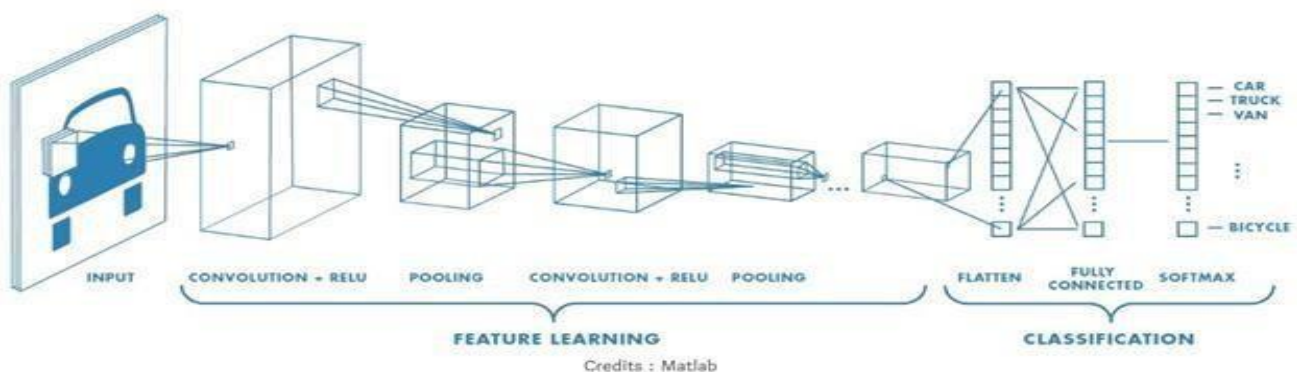
Arsitektur *Neural Network* menggunakan metode *Convolutional Neural Network* sebagai jalur *Artificial Neural Network* (ANN). *Artificial Neural Network* merupakan inti dari proses *Deep Learning* yang



sangat *powerful* untuk melakukan pengklasifikasian citra dengan kompleksitas yang tinggi (Boehmke and Greenwell, 2019). Pada *Deep Learning* dilakukan proses pelatihan dengan dataset citra dan mengubah nilai *pixel* suatu citra dimana proses klasifikasi dapat mendeteksi *patern* dari *input* citra (Chauhan and Ram, 2018).

### 2.1.1. Convolutional Neural Network

*Convolutional Neural Network* (CNN) merupakan pengembangan paling populer digunakan dari *Artificial Neural Network* (ANN), *CNN* umumnya digunakan untuk mengembangkan *neural network* dari data yang bersifat citra/gambar. Arsitektur *CNN* terdiri dari kumpulan *layer-layer* yang akan dilewati *input* citra dalam proses *training*. Dua *layer* pertama (*Convolutional Layer*, *Pooling Layer*) ialah *Convolution Layers* dan diikuti oleh *MaxPooling Layer*, *Layer* ketiga dan keempat ialah *Fully Connected Layer*, dan *Layer* terakhir ialah *output layer* (Lin and Shen, 2018). Gambar 1 menampilkan Arsitektur *CNN* secara umum:



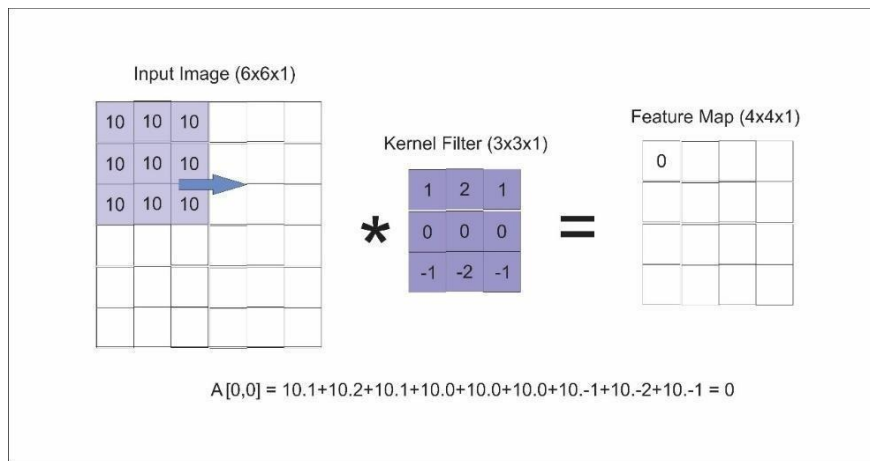
Gambar 1. Arsitektur *CNN*

#### 2.1.1.1. Convolutional Layer

*Convolutional Layer* merupakan bagian layer awal untuk melakukan proses konvolusi. Layer Konvolusi memiliki tujuan untuk mengekstrak *feature* dari *input* mentah gambar. Kumpulan *feature* akan diekstrak dengan cara melakukan sistem perkalian antara matriks dari input gambar dan matriks kernel filter atau *feature* filter. Kernel filter yang berukuran lebih kecil dari matriks input *image*, akan mulai bergerak (*Shift*) dari mulai Matriks input  $[0,0]$  sampai  $[n,n]$ . Formulasi *Convolutional Network* seperti formula (1) (Li *et al.*, 2014):

$$A = f \left( \sum_{l=1}^N I_l * K_{lj} + B_j \right) \quad (1)$$

Setiap matriks input akan dilakukan konvolusi dengan tiap kernel matriks . Setiap hasil konvolusi akan dijumlahkan dengan bias yang akan menghasilkan sebuah fungsi aktivasi sebelum dibentuk ke *feature map*. Gambar 2 menggambarkan proses ekstrak *feature map* dengan kernel filter *Convolutional network*:



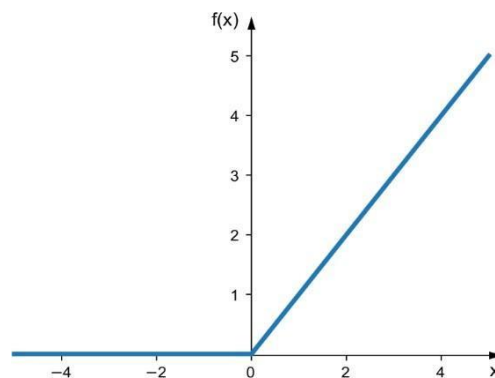
Gambar 2. Proses ekstrak *feature map* dari *convolutional network*

#### 2.1.1.2. Rectified Linear Unit Layer (ReLU)

*Rectified Linear Unit Layer (ReLU)* merupakan jenis fungsi aktivasi yang diada pada *CNN*. *ReLU* pada *CNN* berfungsi untuk melakukan pembersihan nilai negatif pada *layer* dengan nilai 0 ini akan membantu *CNN* mendapatkan *value* yang stabil dan tidak terhenti di sekitaran 0 atau tak terhingga (Sharma, Jain and Mishra, 2018). Fungsi aktivasi *ReLU* dapat diformulasikan seperti formula (2):

$$f(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases} \quad (2)$$

Dimana fungsi akan bernilai  $x$  jika nilai  $x$  lebih dari atau sama dengan 0, dan fungsi akan bernilai 0 jika nilai  $x$  bernilai kurang dari 0. Gambar 3 merupakan tampilan grafik fungsi aktivasi *ReLU*:

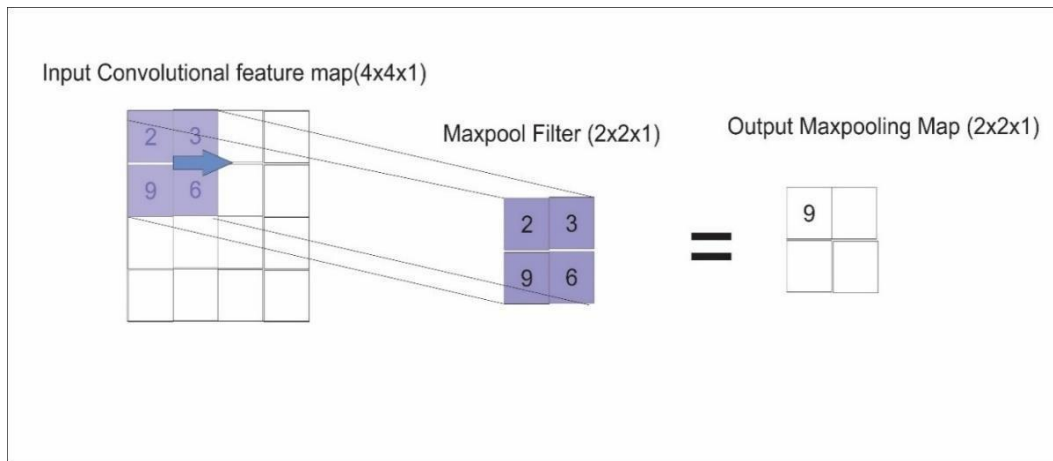


Gambar 3. Grafik fungsi aktivasi *ReLU*

#### 2.1.1.3. Max Pooling Layer

*Max Pooling Layer* adalah sebuah layer lanjutan yang menerima input matriks dari *Convolutional Layer* yang pada layer ini dilakukan proses *subsampling* atau *pooling operation*. *Pooling operation* diterapkan pada tiap *feature map* hasil *Convolutional Layer* yang telah di aktivasi dengan fungsi aktivasi *ReLU*. Cara *Max Pooling Layer* memiliki kesamaan terhadap *Kernel Filter* pada *Convolutional Layer*, *Max Pooling Layer* akan menerapkan *Kernel* matriks baru pada *feature map* kemudian akan

memilih nilai tertinggi dari setiap Kernel matriks melewati *feature map* (*Shift*). Sehingga diakhir proses *Max Pooling* akan menghasilkan dimensi *feature map* yang lebih kecil dari sebelumnya. Gambar 4 menggambarkan proses *Max pooling* pada hasil *feature map* dari *convolutional layer*.



Gambar 4. Proses *max pooling* pada *feature map*

#### 2.1.1.4. Fully Connected Layer

*Fully connected layer* menerima hasil dari *Max Pooling Layer* yang sebelumnya telah dilakukan proses *flatten*. Proses *Flatten* ialah proses meratakan tiap *feature map* yang berbentuk *multidimensional array* menjadi *vector* *n*-dimensi, *n* disini berarti total kelas output yang harus dipilih program. Sebuah *fully connected layer* terdiri dari banyak *layer*, dan tiap *layer* memiliki banyak neuron yang melakukan proses perkalian antara input matriks dan bobot masing-masing input. Proses perkalian diformulasikan seperti formula (3):

$$u_t = f\left(\sum_{i=1}^N w_{t,i}x_t + b_f\right) \quad (3)$$

Dimana hasil fungsi aktivasi dari nilai perkalian input matrix dan bobot. Dan hasilnya di jumlahkan dengan bias yang ada.

#### 2.1.1.5. Softmax

*Softmax* adalah langkah akhir dari rangkaian Arsitektur *CNN*, *Softmax* berfungsi sebagai mengembalikan nilai probabilitas terbesar ke dalam kelas- kelas yang tepat.

## 2.2. Perancangan Sistem

### 2.2.1. Persiapan Dataset

Dalam melaksanakan penelitian ini, persiapan dataset merupakan langkah awal untuk memulai pembuatan sistem. Dataset dalam penelitian ini berupa kumpulan gambar daun yang diambil secara

manual menggunakan kamera belakang *smartphone* Asus Zenfone Max Pro M1 dengan kualitas kamera 13MP. Dataset diambil dengan cara mengambil daun herbal yang sehat dari batang pohon, dan diletakan diatas kertas putih. Tiap foto hanya ada 1 jenis daun herbal dan diletakan diatas kertas putih. Hal ini bertujuan untuk mempermudah proses *training* dan meningkatkan akurasi prediksi. Tabel 1 menunjukan jenis daun herbal yang digunakan sebagai dataset.

Tabel 1. Daftar Jenis Daun Herbal

Jenis Daun	Train Data	Validation Data	Test Data
Jambu Biji	41	10	10
Ketela	30	10	10
Lidah Buaya	30	10	10
Meniran	30	10	10
Sirih	30	10	10
Tentir	30	10	10
Jumlah	191	60	60

### 2.2.2. Pembuatan *Neural Network*

Dalam Penelitian ini , *Neural Network* dibuat dalam beberapa bentuk untuk menganalisa perbedaan tiap *training*. *Neural Network* menggunakan arsitektur *CNN* dengan menggunakan *Tensorflow + Keras Library*, dan di tulis dengan Bahasa pemrograman *python*. Terdapat 4 jenis *Neural Network* yang disusun dalam bentuk *CNN*. Tabel 2 menunjukan susunan *Neural Network* tanpa tambahan apapun.

Tabel 2. Susunan Layer *CNN basic*

Layer	Size	Output Shape
Input	3x224x224	-
Convolution 1 +ReLU	64 (5 x 5)	64x220x220
Convolution 2 +ReLU	64 (5 x 5)	64x216x216
Maxpooling 1	(2 x 2)	64x108x108
Convolution 3+ReLU	128 (5 x 5)	128x104x104
Convolution 4+ReLU	128 (5 x 5)	128x100x100
Dense	128 neuron	128 neuron
Dense 1	6 kelas	6 kelas

Susunan kedua masih menggunakan susunan *layer* seperti pada Tabel 2, namun pada susunan kedua dilakukan proses data *augmentation*. Data *augmentation* ialah ketika tiap *batch* yang dihasilkan diubah oleh transformasi untuk mengurangi *overfitting* (Wick and Puppe, 2017). Data tiap citra daun herbal akan dilakukan *Rotating* sebesar 15 dan dilakukan *shifting width* sebesar 0.2 dan *shifting height* 0.2. Tabel 3 menunjukan susunan tiap *layer CNN* dengan Data *Augmentation*.

Tabel 3. Susunan Layer CNN + Data Augmentation

Layer	Size	Output Shape
Input	3x224x224	-
Convolution 1 +ReLU	64 (5 x 5)	64x219x219
Convolution 2 +ReLU	64 (5 x 5)	64x214x214
Maxpooling 1	(2 x 2)	64x107x107
Convolution 3+ReLU	128 (5 x 5)	128x102x102
Convolution 4+ReLU	128 (5 x 5)	128x97x97
Dense	128 neuron	128 neuron
Dense 1	6 kelas	6 kelas

Susunan Ketiga menggunakan *transfer learning*. *Transfer Learning* memungkinkan peneliti tidak menulis seluruh layer dari awal. Pada penelitian ini peneliti menggunakan VGG16 sebagai *feature extraction* dengan menggunakan *imagenet* sebagai *weight* Tabel 4 menunjukkan susunan tiap layer dengan VGG16 sebagai *feature extraction*.

Tabel 4. Susunan Layer VGG16 sebagai Feature Extraction

Layer	Size	Output Shape
Input	3x224x224	-
VGG16	(pretrained)	64x219x219
Dense + ReLu	256 neuron	256 neuron
Dense + Softmax	6 kelas	6 Kelas

Susunan Keempat menggunakan susunan yang sama seperti Tabel 4, namun dilakukan proses *finetuning*. Yaitu dengan memperkecil *learning rate*. Tabel 5 menunjukkan susunan tiap layer dengan *Finetuning VGG16*.

Tabel 5. Susunan Layer Finetuning VGG16

Layer	Size	Output Shape
Input	3x224x224	-
VGG16	(pretrained)	64x219x219
Dense + ReLu	256 neuron	256 neuron
Dense + Softmax	6 kelas	6 Kelas

### 2.2.3. Data Preprocessing

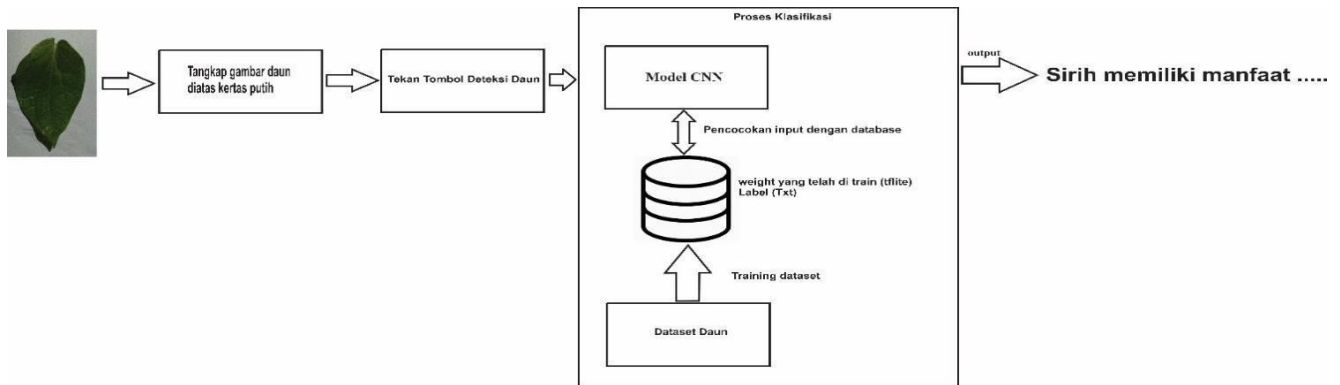
Data yang telah disiapkan masih dalam multi-*pixel*, yang artinya akan mengakibatkan proses *training* menjadi tidak maksimal dan mengurangi tingkat akurasi. Maka sebelum dilakukan *training*, diperlukan proses *resize image* sehingga gambar yang dilatih memiliki ukuran *pixel* yang sama. Dalam penelitian ini peneliti mengubah ukuran image menjadi 224x224 *pixel*.

### 2.2.4. Proses Training

Langkah selanjutnya setelah dataset dan *Neural Network* telah siap, maka dilakukan proses *training* dengan masing-masing susunan CNN terdiri dari 200 perulangan dengan *learning rate* 0.001 kecuali susunan CNN keempat seperti tergambar pada tabel 5. Output proses *training* berupa *file h5* yang

merupakan *trained graph*. Untuk dilakukan *deploy* ke sistem android diperlukan konversi *file h5* ke *file tensorflow lite*.

### 2.2.5. Perancangan sistem *Android*



Gambar 5. Arsitektur sistem *Android*

Gambar 5 menampilkan arsitektur sistem *Android*, diawali dari pengambilan citra daun dan diletakan diatas kertas putih. Ketika citra daun ditangkap oleh kamera *Android*, citra daun langsung di masukan ke proses klasifikasi yang melibatkan *trained graph* dan label. Proses klasifikasi mengeluarkan output label dengan tambahan *string* manfaat singkat tentang citra daun yang telah diproses.

## 3. HASIL DAN PEMBAHASAN

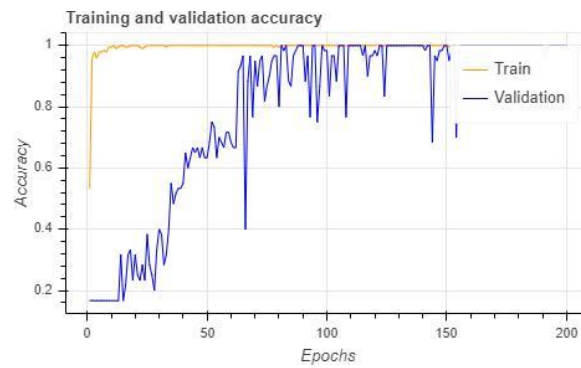
### 3.1. Persiapan Pengujian

Persiapan Pengujian dibutuhkan 4 jenis neural network yang terbentuk, 2 *neural network CNN* dan *VGG16* yang masing-masing diberi tambahan 2 algoritma lain seperti *data augmentation* dan *Finetunning*. Pengujian *Dataset* daun membutuhkan *Google Colab* untuk menulis *neural network* yang nantinya dilakukan pelatihan terhadap *Dataset* yang telah disiapkan. Penggunaan *Google Colab* dikarenakan untuk mendapatkan performa *training* yang lebih cepat dibandingkan dilakukan *training* secara *offline*.

### 3.2. Hasil Pengujian *Neural Network*

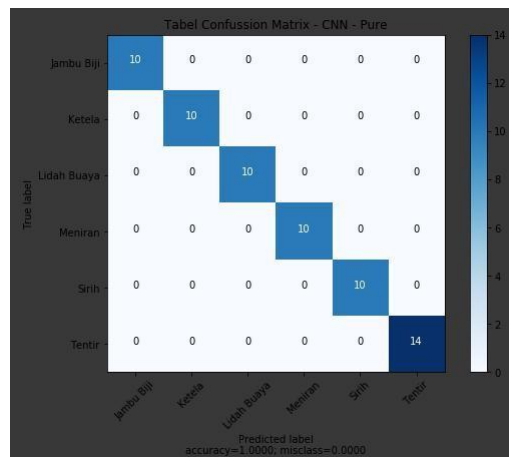
Pengujian *Neural Network* dilakukan setelah proses *training* dataset telah mencapai tahap akhir. Tahap ini digunakan untuk menguji tingkat efisiensi *Neural Network* dalam melakukan *training* dataset. Tingkat Keberhasilan suatu *Neural Network* dalam melakukan *training* dapat dilihat dari tingkat akurasi *training*. Tingkat akurasi merupakan pengukuran performa paling intuitif (Barbosa *et al.*, 2017). Selain Tingkat akurasi, Tingkat *Loss* juga akan berpengaruh untuk penilaian kelayakan *Neural Network*.

### 3.2.1. Pengujian Training dengan CNN



Gambar 6. Hasil akurasi *training CNN*

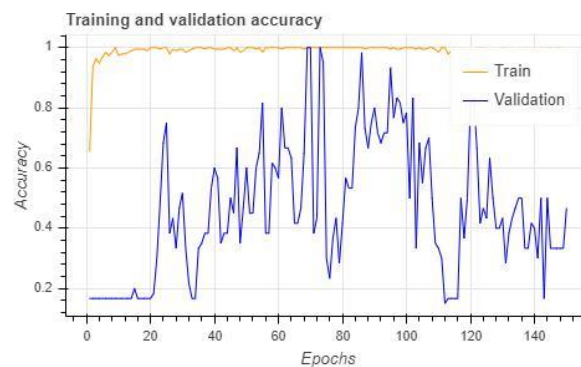
Gambar 6 menunjukkan hasil akurasi *training* dengan menggunakan *CNN* dengan 200 *epoch* dan *learning rate* 0.001, grafik menunjukkan kenaikan yang tidak stabil diproses validasi meski pada proses *training* cukup baik menunjukkan kelayakan *neural network*.



Gambar 7. Hasil Plot *Graph CNN* Ke *Test Data*

Gambar 7 *Plotting graph* untuk mencoba melakukan klasifikasi terhadap data baru diluar data *train* dan data validasi. *Plotting graph* memang menunjukkan akurasi prediksi setiap kelas secara sempurna. Namun pada label kelas “Tentir” terbaca 14 total data dari 10 data, yang berarti ada masalah dalam pengklasifikasian terhadap label dari data diluar data *train* dan data validasi

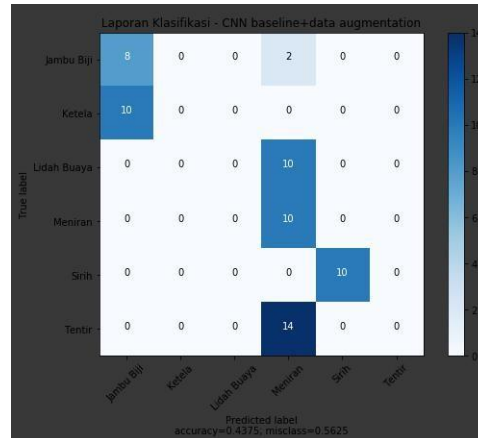
### 3.2.2. Pengujian Training dengan CNN + Data Augmentation



Gambar 8. Hasil akurasi *training CNN + Data Augmentation*

Gambar 8 menunjukkan hasil akurasi proses training dengan penambahan *Data Augmentation*.

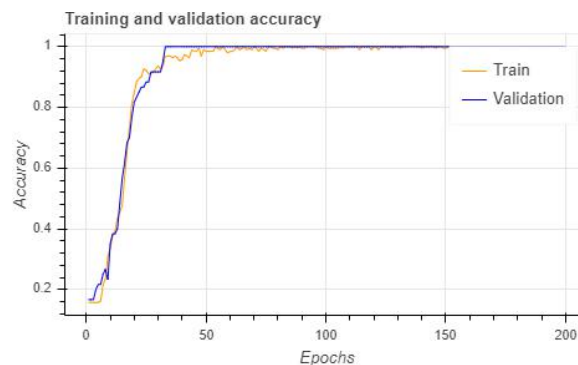
Penambahan Data *Augmentation* dilakukan karena pada Pengujian 3.1.1 terdapat *missclass* saat *plotting graph* ke data *test*. Penerapan Data *Augmentation* yang dilakukan meliputi perotasian gambar, *stretch* gambar, pembalikan posisi gambar. Terlihat di grafik validasi banyak terjadi loncatan akurasi dari 0.2 sampai 0.9. Grafik tidak menunjukan aktifitas yang lebih baik dari Gambar 6.



Gambar 9. Hasil plot *Graph CNN+Data Augmentation* Ke *test* data

Gambar 9 menunjukkan Hasil *plotting graph CNN + Data Augmentation* ke data *test*. Terlihat lebih banyak kesalahan dari Gambar 10. Terjadi *missclass* pada Label “Meniran” dan semua label tidak menunjukkan jumlah total data yang seharusnya. Tingkat akurasi juga menurun menjadi 0.4 dari Gambar 7 yang menunjukkan tingkat akurasi 1.0. Sehingga bisa dinyatakan bahwa *Neural Network* pada pengujian *CNN+Data Augmentation* tidak berhasil.

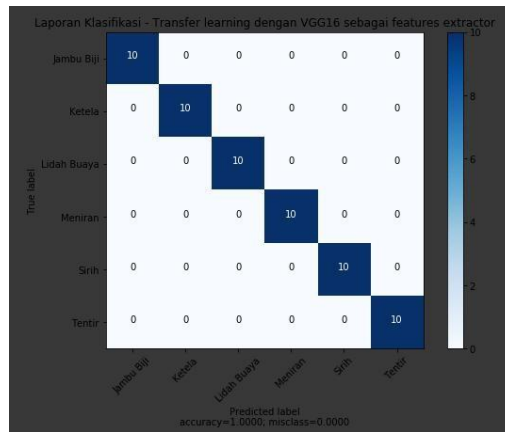
### 3.2.3. Pengujian Training dengan *Transfer Learning* (VGG16 sebagai *Feature Extraction*)



Gambar 10. Hasil akurasi *training transfer learning* (VGG16 sebagai *feature extraction*)

Gambar 10 menunjukkan hasil akurasi *training transfer learning* dengan VGG16 sebagai *feature extraction*. Grafik menunjukkan kenaikan yang stabil baik pada grafik *train* maupun validasi. Grafik validasi terjadi kestabilan nilai akurasi di *epoch* 40 sedangkan untuk grafik *train* terjadi kestabilan nilai akurasi di *epoch* 80.

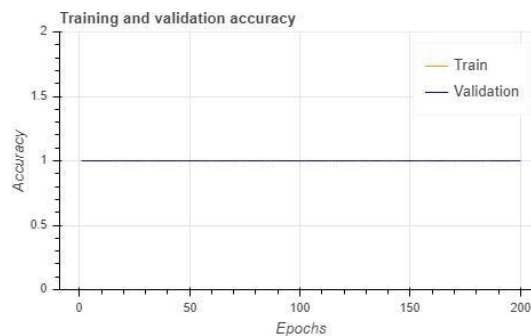




Gambar 11. Hasil plot *graph training transfer learning* (VGG16 sebagai *feature extraction*) Ke *test data*

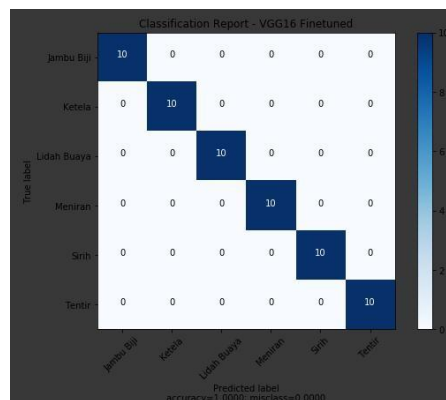
Gambar 11 menunjukkan hasil *plotting Graph transfer learning* dengan VGG16 sebagai *feature extraction* ke data *test*. *Plotting* menunjukkan akurasi 1.0 dan tidak ada *missclass* pada setiap label. Terlihat perbedaan pada label “Tentir”, pada Gambar 7 terjadi kesalahan jumlah data pada Gambar 13 *neural network* berhasil melakukan pengklasifikasian dengan baik.

### 3.2.4. Pengujian Training dengan VGG16 + *Finetunning*



Gambar 12. Hasil akurasi *training transfer learning* (VGG16+*Finetunning*)

Gambar 12 menunjukkan hasil akurasi *training transfer learning* dengan VGG16+*Finetunning*. *Finetunning* dalam penelitian ini hanya mengganti nilai *learning rate* yang sebelumnya 0.001 menjadi 0.00001 dengan harapan mendapatkan hasil yang lebih optimal dalam akurasi. Terlihat grafik *train* dan validasi masing-masing stabil di angka 1 dari awal *epoch* sampai akhir *epoch*.



Gambar 13. Hasil plot *graph training transfer learning* (VGG16 + *Finetunning*) ke *test data*

Gambar 13 menunjukkan hasil *plotting graph training transfer learning* dengan *VGG16+Finetunning* ke *test data*. *Plotting* memperlihatkan keberhasilan neural network mengklasifikasikan dengan tepat tiap data pada *test data* kedalam kelas yang tepat.

### 3.3. Analisa Pengujian Neural Network

Pengujian *neural network* dilakukan dengan mengukur akurasi *train* dan akurasi validasi tiap model yang telah dipakai sebagai dasar penyusunan *neural network*.

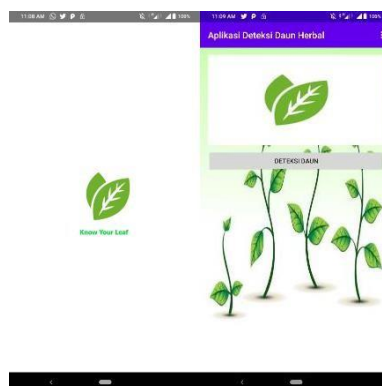
Tabel 6. Analisa pengujian *neural network*

Dataset Train	Epoch	Model	Akurasi train	Akurasi Validasi
191 data	200 epoch	CNN	100%	100%
		CNN + Data Augmentation	44%	47%
		VGG16	100%	100%
		VGG16 + Finetunning	100%	100%

Dari Tabel 6 dengan melakukan *train* terhadap 191 dataset dengan jumlah epoch sebesar 200. Model *CNN* mampu mencapai nilai sempurna untuk akurasi *train* sebesar 100% dan akurasi validasi sebesar 100%, dalam proses *plotting* seperti Gambar 10 menampilkan ada masalah “*missclass*” meski tidak terdeteksi. Model *CNN + Data Augmentation* memiliki performa *training* lebih buruk dari model *CNN* biasa, untuk akurasi *train* hanya memperoleh 44% sedangkan akurasi validasi memperoleh nilai 47%. Hal ini tentu akan mempengaruhi performa *klasifikasi* seperti yang tertera di Gambar 13. Model *VGG16* memiliki nilai keberhasilan yang sama dengan model *CNN*, untuk akurasi *train* mencapai nilai 100% sedangkan akurasi validasi mencapai 100%. *VGG16* lebih berhasil dalam *plotting* seperti di Gambar 16, jumlah *plotting* sama dengan jumlah data tiap kelas. *VGG16+Finetunning* menghasilkan nilai akurasi *train* 100% dan akurasi validasi 100%, secara grafik *train* dan validasi seperti Gambar 17 bahwa penambahan *finetunning* yang berarti mengubah *learning rate* menjadi 0.00001.

### 3.4. Hasil Sistem Android

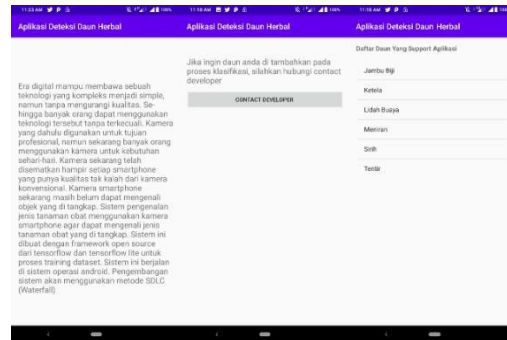
#### 3.4.1. Tampilan *SplashScreen* dan Menu Utama



Gambar 14. Tampilan *Splashscreen* dan menu utama

Gambar 14 menunjukkan tampilan *splashscreen* dan menu utama. Tampilan *splashscreen* muncul diawal ketika aplikasi dibuka, Tampilan menu utama akan muncul setelah 2 detik tampilan *splashscreen*. Tampilan menu utama hanya ada tombol “Deteksi Daun” jika diklik maka menuju halaman klasifikasi.

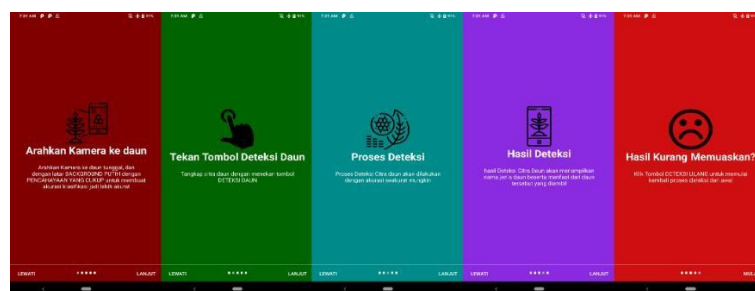
3.4.2. Tampilan Opsi Menu Utama



Gambar 15. Tampilan opsi menu utama (opsi tentang, kontak, daftatr daun yang *support* aplikasi)

Gambar 15 menampilkan opsi menu *dropdown* pada menu utama. Menu *dropdown* secara *default* tidak muncul tanpa di klik sebelumnya. Isi dari opsi menu utama yaitu opsi tentang yang menampilkan abstraksi Penelitian, opsi kontak yang berisi sebuah petunjuk untuk mengontak *developer* jika ada yang ditanyakan, opsi terakhir yaitu daftar daun yang *support* aplikasi berisi macam-macam jenis daun yang di-*support* oleh aplikasi.

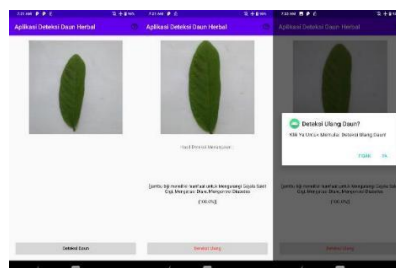
### 3.4.3. Tampilan Menu *Help* di Laman Klasifikasi



Gambar 16. Tampilan Petunjuk di Laman Klasifikasi

Gambar 16 menunjukkan 4 *slide* tentang petunjuk memulai deteksi daun. Diawali dari *capture* daun, proses deteksi, hingga hasil ditampilkan dilayar.

### 3.4.4. Tampilan Laman Klasifikasi



Gambar 17. Tampilan menu klasifikasi

Gambar 17 menunjukkan tampilan awal menu klasifikasi. Didalam menu klasifikasi terdapat kamera *preview* untuk *capture* citra daun, kemudian tombol “Deteksi Daun” untuk mengambil gambar untuk dilakukan klasifikasi jenis. Selama proses klasifikasi tombol “Deteksi Daun” berubah menjadi tombol “Deteksi Ulang”. Saat proses Klasifikasi selesai maka akan ada tampilan hasil klasifikasi dibagian bawah kamera *preview*. Tombol “Deteksi Ulang” diklik maka akan menampilkan *alert* sebagai peringatan untuk mengulangi lagi langkah klasifikasi daun.

## 4. PENUTUP

### 4.1. Kesimpulan

Penelitian ini mampu mengaplikasikan hasil training dengan *Deep Learning* dengan berbagai metode *CNN* kedalam sebuah aplikasi pengenalan tanaman obat. Hasil *training neural network* menghasilkan hasil dengan akurasi 100% dengan total *epoch* 250. Hasil akurasi 100% dapat dicapai karena peneliti menghilangkan halangan pada dataset yang menyebabkan terjadinya penurunan akurasi *train* maupun validasi, namun terjadi perubahan akurasi menjadi 43% Ketika dataset diubah bentuk dengan menggunakan data *augmentation*. Halangan yang dihilangkan pada dataset meliputi penghapusan warna *background*, gambar yang difoto dalam keadaan penuh tanpa ada potongan, posisi gambar yang di samaratakan.

### 4.2. Saran

Diharapkan untuk pengembangan selanjutnya dengan memperbanyak jumlah *support* daun yang lebih umum, dan penggunaan arsitektur *neural network* yang lain untuk memperakurat proses klasifikasi.

## DAFTAR PUSTAKA

- Barbosa, J. B. *et al.* (2017) ‘HerbApp : A Mobile-Based Application for Herbal Leaf Recognition Using Image Processing and Regularized Logistic Regression Classifier’, 2(10).
- Boehmke, B. and Greenwell, B. (2019) *Hands-On Machine Learning with R, Hands-On Machine Learning with R*. doi: 10.1201/9780367816377.
- Chauhan, K. and Ram, S. (2018) ‘International Journal of Advance Engineering and Research Image Classification with Deep Learning and Comparison between Different Convolutional Neural Network Structures using Tensorflow and Keras’, pp. 533–538.
- Gifary, S. (2015) ‘Intensitas Penggunaan Smartphone Terhadap Perilaku Komunikasi Sharen’, *Jurnal Sosioteknologi*, 14(2), pp. 170–178. Available at: <http://journals.itb.ac.id/index.php/sostek/article/view/1472/1045>.
- Li, Q. *et al.* (2014) ‘Medical image classification with convolutional neural network’, *2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014*, 2014(December), pp. 844–848. doi: 10.1109/ICARCV.2014.7064414.

- Lin, G. and Shen, W. (2018) 'Research on convolutional neural network based on improved Relu piecewise activation function', *Procedia Computer Science*. Elsevier B.V., 131, pp. 977–984. doi: 10.1016/j.procs.2018.04.239.
- Ni'mah, F. S., Sutojo, T. and Setiadi, D. R. I. M. (2018) 'Identifikasi Tumbuhan Obat Herbal Berdasarkan Citra Daun Menggunakan Algoritma Gray Level Co-occurrence Matrix dan K-Nearest Neighbor', *Jurnal Teknologi dan Sistem Komputer*, 6(2), p. 51. doi: 10.14710/jtsiskom.6.2.2018.51-56.
- S. Ifandi, Jumari, dan S. Widodo AS, "Keanekaragaman Jenis Tumbuhan Obat Masyarakat Suku Kaili di Dusun Tompu Kecamatan Sigi Biromaru Kabupaten Sigi Sulawesi Tengah," dalam Seminar Nasional Biologi II, Semarang, 2015.
- Santoso, A. and Ariyanto, G. (2018) 'Implementasi Deep Learning Berbasis Tensorflow', *Jurnal Emitter*, 18(01), pp. 22–27.
- Sari, I. D. *et al.* (2015) 'Tradisi Masyarakat dalam Penanaman dan Pemanfaatan Tumbuhan Obat Lekat di Pekarangan', *Jurnal Kefarmasian Indonesia*, 5(2), pp. 123–132. doi: 10.22435/jki.v5i2.4407.123-132.
- Sharma, N., Jain, V. and Mishra, A. (2018) 'An Analysis of Convolutional Neural Networks for Image Classification', *Procedia Computer Science*. Elsevier B.V., 132(Iccids), pp. 377–384. doi: 10.1016/j.procs.2018.05.198.
- Wick, C. and Puppe, F. (2017) 'Leaf Identification Using a Deep Convolutional Neural Network'. Available at: <http://arxiv.org/abs/1712.00967>.